Stochastic Time-Domain Mapping for Comprehensive Uncertainty Assessment in Eye Diagrams

Stochastic Time-Domain Mapping for Comprehensive Uncertainty Assessment in Eye Diagrams

Mihai Telescu, *Member, IEEE*, Riccardo Trinchero, *Member, IEEE*, Nastaran Soleimani, *Student Member, IEEE*, Noel Tanguy, and Igor Simone Stievano, *Senior Member, IEEE*

***Abstract*—The eye diagram is one of the most common tools used for quality assessment in high-speed links. This article proposes a method of predicting the shape of the inner eye for a link subject to uncertainties. The approach relies on machine learning regression and is tested on the very challenging example of flexible link for smart textiles. Several sources of uncertainties are taken into account related to both manufacturing tolerances and physical deformation. The resulting model is fast and accurate. It is also extremely versatile: rather than focusing on a specific metric derived from the eye diagram, its aim is to fully reconstruct the inner eye and enable designers to use it as they see fit. This article investigates the features and convergence of three alternative machine learning algorithms, including the single-output support vector machine regression, together with its least squares variant, and the vector-valued kernel ridge regression. The latter method is arguably the most promising, resulting in an accurate, fast and robust tool enabling a complete parametric stochastic map of the eye.**

***Index Terms*—Eye diagram, high-speed interconnects, kernel machine regressions, signal integrity, smart textiles, uncertainty quantification, wearable electronics.**

I. Introduction

BENEFITING from constant interest from both industry and academia, stochastic analysis evolved significantly over the past years. Various techniques are currently available, allowing designers to evaluate the impact of system uncertainties for various circuit applications, packaging, and interconnects. Historically, the most common approach to stochastic analysis was Monte Carlo (MC). Featuring several improved versions, the technique is straightforward. Data are collected via repeated simulation and allows various metrics to be computed, quantifying the performance and robustness of a system subject to uncertainties related either to the manufacturing process or environment parameters [1], [2]. The main drawback of MC is its computational cost. This prompted researchers to look for alternative solutions to the computationally expensive brute-force MC sampling, allowing them to infer the statistical characteristics of a system from limited data. Many of the first successful approaches were based on polynomial chaos expansion (for an overview of this family of algorithms see [3]). These were followed by other machine learning inspired solutions falling mainly into the category of kernel machine regression (e.g., [4], [5], [6], [7], [8], [9]) and artificial neural networks (ANNs) (e.g., [10], [11], [12], [13], [14]).

This article focuses on kernel machine regression techniques, such as: the support vector machine (SVM) regression [15], the least-squares support vector machine (LS-SVM) [16] regression, and the more recent vector-valued kernel ridge regression (KRR) [6], [7], [17], which have been proven particularly effective for various microelectronics and radio-frequency applications [4], [5], [8]. Indeed, kernel machine regressions provide an interesting alternative to the otherwise more flexible ANNs, especially well-suited for problems in which a "relatively small" set of training data is available [15], [16], [18]. Unlike ANNs, the linear model structure adopted by kernel regression (i.e., the model unknowns appear linearly) has the key advantage of heavily simplifying the training phase, which reduces to the solution of a standard convex optimization problem, thus leading to several advantages in terms of training time and accuracy w.r.t. the number of training samples [6], [10].

In this specific context, this article explores a new methodology for uncertainty quantification in the field of signal integrity. The inner boundaries of an eye diagram are approximated by a polygon. Several kernel machine regression algorithms are then used to build a surrogate model capable of mimicking the impact of system uncertainties on this polygon, resulting in a "stochastic map." A particularly challenging example was selected in order to demonstrate the technique: a flexible differential digital link for smart textiles. The implementation uses copper yarn and is subject to various sources of uncertainties, including high tolerances during manufacturing [19]. It is important to observe that unlike previous approaches, which predicted the impact of uncertainties on specific metrics derived from the eye diagram (e.g., vertical opening, eye SNR, etc), stochastic time-domain

Manuscript received 17 April 2023; revised 20 July 2023; accepted 16 September 2023. Date of publication 4 October 2023; date of current version 13 December 2023. *(Corresponding author: Riccardo Trinchero.)*

Mihai Telescu and Noel Tanguy are with the University of Brest, CNRS, Lab-STICC, F-29238 Brest, France (e-mail: mihai.telescu@univ-brest.fr; noel.tanguy@univ-brest.fr).

Riccardo Trinchero, Nastaran Soleimani, and Igor Simone Stievano are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy (e-mail: riccardo.trinchero@polito.it; nastaran.soleimani@polito.it; igor.stievano@polito.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TEMC.2023.3317974.

Digital Object Identifier 10.1109/TEMC.2023.3317974

mapping fully models inner-eye geometry. Several metrics may thus be computed in postprocessing, the use of masks is also possible. The aim is to allow a statistically meaningful design leading to more robust systems. The approach is particularly interesting for cases where high tolerances are an issue in the design flow. These include emerging technologies with nonstandard manufacturing processes, or low-cost products for mass consumers where high tolerances are a result of financial considerations. Note, however, that the stochastic mapping approach is not application-oriented but general and very adaptable.

The rest of this article is organized as follows. An example related to wearable electronics, based on published data, is presented in Section II. A description of how the eye diagram is processed follows in Section III. Section IV provides an in-depth description of the algorithms used to perform stochastic mapping. Numerical results relative to the selected example are collected in Section V. Finally, Section VI concludes this article.

II. Test Case

In the past two decades significant progress has been made in the field of wearable electronics and wearable computing. Fabrics integrating sensors, processors, energy harvesters, lights and batteries, so called "smart textiles" or "e-textiles," have attracted the interest of various private and public actors [19], [20], [21], [22], [23], [24], [25]. The defense industry has a particular interest for this field: providing the soldier on the modern battlefield with apparel integrating sensor hubs, processors, and various interfaces has numerous advantages [21]. Civilian applications also exist and are either health-oriented or focus on various specialized garments designed for exploration, hazardous environments, or extreme sports. In a more general context, e-textiles may also be used in nonwearable applications (e.g., smart bedsheets) or even in applications, which are not human-oriented (geotextiles) [24].

This new generation of electronics brings new challenges in terms of signal and power integrity. While the main design objective is always the transmitting electrical signals with as little distortion as possible, the specific context of e-textiles magnifies the impact of mechanical factors and forces designers to carefully consider aspects, such as environment stability and service life [23]. Flexible data busses are characterized by much higher manufacturing imperfections than the ones in regular PCBs and suffer greater impact from environment parameters, such as temperature and humidity. External mechanical action, both in terms of compression and stretching is also an issue.

The results obtained in this article are based on the serial link, as shown in Fig. 1, which has been implemented and simulated in LTSPICE [26]. It consists of a differential data communication channel implemented by conductive copper wires weaved in a textile structure [27]. The flexible interconnect is described using the classic multiconductor transmission line theory and the per unit length parameters computed from the electrical and geometrical information of the structure [28], [29]. Note that conductive fibers remain, as of 2023, the most popular technological solution for interconnects in e-textiles [30]. The link features a differential driver at the near end and its paired

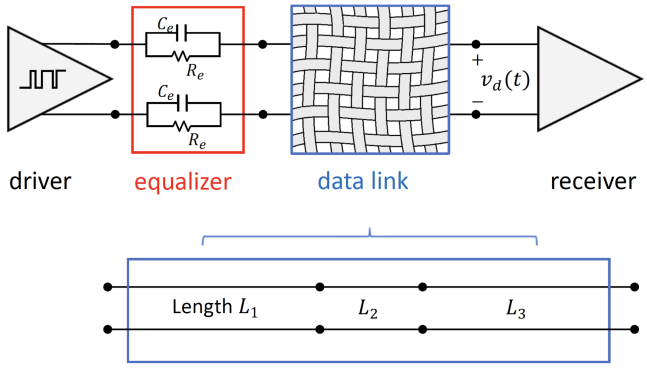

Fig. 1. System under test: digital link for wearable applications. The nominal values of the equalizer and of the line lengths are: $R_e = 70\,\Omega$, $C_e = 3.2\,\text{pF}$, $L_1 + L_2 + L_3 = 33\,\text{cm}$.

receiver at the far end. In order to increase maximum data rates, a passive equalizer is inserted between the driver and the line. It is assumed that the equalizer and buffers are implemented either using conventional electronics or semiflexible substrates. They are affected by uncertainties, albeit to a lesser extent. The center section of the interconnect (identified by a transmission line segment with length L_2 in the scheme) accounts for a potentially altered geometry. This can either take the form of compression due to external mechanical action or to the very nature of the application (e.g., bending at knees or elbows) or on the contrary stretching of the material. Either situation alters the distance between the conductors on a specific segment of the link. The circuit model accounts for the resulting discontinuity.

NRZ encoding is used at a data rate of 2 Gbps. At this stage of the investigation, the driver and the receiver are modeled using compact, linear models, very similar to the ones implemented in IBIS-AMI simulations. Specifically, an equivalent Norton-based voltage controlled circuital representation of the differential drivers and receivers is used, according to the method developed in [31]. This representation enables the accurate simulation of coupled channels including both differential- and common-mode signals. The driver and the receiver currents are defined as a function of the pertinent output and input port voltages, respectively, via a constant conductance matrix. The dynamic behavior of the devices is represented by a dynamic linear time-invariant submodel. Time-varying current source terms are instead used to represent the switching activity of the driver in terms of a time varying analog waveform. Device parameters are representative of the application at hand and the involved data rate. For a detailed presentation of the buffer models see (2) and Fig. 2 in [31] and the explanations therein.

However, the methodology described in this article can be used as is with nonlinear buffer models. For applications where such models are required for increased accuracy, they may readily be used [31], [32], [33]. It is important to note that the geometrical specifications of the interconnect, including the fabrication tolerances, are based on the experimental work published in [19]. The selected structure is labeled as GSSG-1 (see Fig. 3 in the previously referenced paper) and consists in a differential pair between two ground conductors.

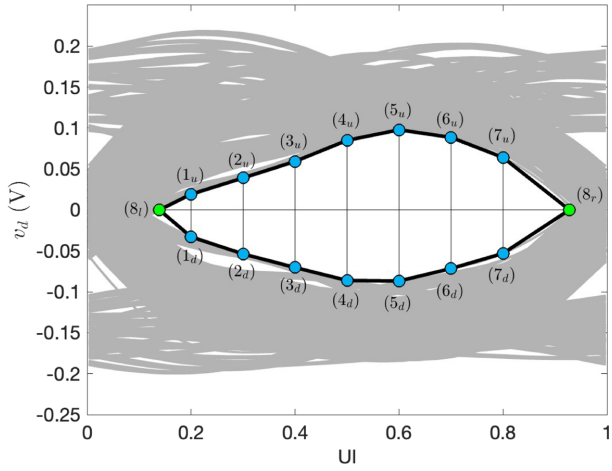

Fig. 2. Example eye diagram obtained for the data link of Fig. 1 for a given set of uncertain parameters. The plot also collects the information about the location of the polygonal edges associated to inner-eye aperture, where the eye vertical apertures are labeled through the pairs from 1 to 7 and the eye width is indicated via the label pair number 8.

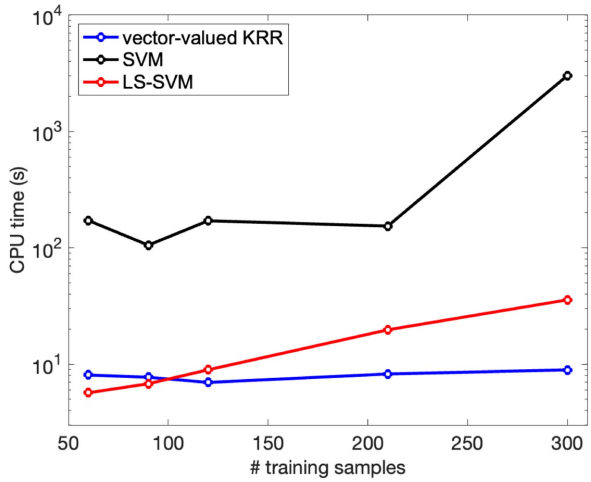

Fig. 3. Comparison of the training time of the proposed regression techniques for an increasing number of training samples. The CPU times do not include the computational cost for the generation of the training set.

Without loss of generality, in our application, 11 uniformly distributed uncertain parameters are considered. The distance between the signal lines is 411 μm $\pm$ 7.3%. The distance between each signal line and the adjacent ground line is 481 μm $\pm$ 6.7%. Conductor diameter is 228 μm $\pm$ 11.1%. The total length of the link is 33 cm $\pm$ 1 cm (mainly due to sizing). The section under mechanical stress varies in length from 2 to 5 cm and the compression [stretching] factor is 25% [10%]. Tolerances of 1% are assumed for the elements of the equalizer. Each eye diagram is computed for 4000 bits.

III. Eye Aperture and Data-Link Quality

Throughout this article, the performance of the communication via the data link is quantitatively assessed using the eye aperture on the receiver side, as obtained by observing the differential voltage response $v_d(t)$.

It is common practice to use the eye diagram when designing high-speed interconnect systems. Associating eye-diagram analysis to uncertainty quantification and stochastic analysis has been investigated in recent years [34], [35]. In addition, thorough theoretical research has consolidated eye computation and processing, as well as jitter analysis making them highly reliable, standard tools [36], [37]. Machine learning techniques, such as the ones used in this article, have also been used in the past in relation to eye-diagram analysis [5], [38], [39]. The novelty of the present work comes from the way in which machine learning is used, from the metamodeling strategy itself. Instead of focusing on specific parameters derived from the eye diagram and computed as scalar quantities (eye height, horizontal opening, etc.), the clean inner eye is literally mapped by a polygonal representation. The metamodeling process focuses on the vertices of the polygon. With this approach, most of the metrics commonly used in data-link design can be readily computed a posteriori, in a stochastic framework.

It should be noted that the methodology described in this article may be used regardless of the way in which the eye pattern is obtained. Postprocessing the image of the eye is one option, using sampled recorded responses is another. The former alternative usually relies on both in-house or commercial solutions enabling the fast simulation of a very large number of bits (on the order of millions) [31], [36]. The latter is usually preferred when the problem has a dominant nonlinear nature, usually due to the buffers. In this case, a classic transient simulation is carried out by means of a circuit solver, such as SPICE, where transceivers are replaced by physics-based transistor-level models or by faster behavioral nonlinear surrogates, yielding a sampled transient response of the pertinent variable of interest (e.g., $v_d(t)$) [33]. The inherent limitation in this case is the CPU time required to run SPICE and usually this is feasible when a relatively limited number of bits is considered, at most on the order of tens or hundreds of thousands.

A possible implementation of a compact routine allowing to compute the polygonal shape of the aperture and parameters, such as the eye area is detailed in the appendix. This pseudocode is made available in order to provide readers with a fully documented and self-contained set of tools for both the computation of the inner eye, as described in this section and the surrogate modeling, as described in the next section.

Fig. 2 shows an example eye diagram, which can be obtained for the considered test case, together with its inscribed inner polygon (see the dark thick line) obtained with the proposed procedure. The diagram is generated from the data-link response computed for a given sample of the uncertain parameters. The blue circles in the figure are the minimum and the maximum vertical positions of the clean inner-eye area observed at a given set of points in the normalized unit interval. The green circles, instead, represent the jitter and can be used to define the horizontal eye aperture arising from the zero-crossing points, leading to the left and right vertices of the inner polygon.

It is important to point out that the optimal number of vertices of the inner polygon shown in Fig. 2 depends on the shape of the eye and thus is strongly related to the specific application at hand. In this article, the points have been selected to guarantee a good accuracy of the approximated eye area. If needed, an

automatic procedure can be devised, based on a heuristic step-by-step greedy algorithm. Basically, the number of points along the abscissa, which represent the horizontal coordinates of the vertices, would gradually be increased in the range [0, 1] of the unit interval until the variation of the resulting eye area becomes negligible. A relative threshold mechanism can be set. Also, one should avoid vertical probing in the area affected by time-domain jitter (e.g., the UI points 0.1 or 0.9 in the example eye). As already mentioned, horizontal probing is used to model this type of uncertainty.

IV. Surrogate Models

A. Problem Statement

Let us consider the problem of approximating the variability of the vertices of a polygon approximating a generic eye diagram collected in the vector $\boldsymbol{y} = [y_1, \ldots, y_D]^T \in \mathcal{Y}$ with $\mathcal{Y} \subseteq \mathbb{R}^D$ as a function of a set of parameters of the link collected in the vector $\boldsymbol{x} = [x_1, \ldots, x_p]^T \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^p$. For the example eye of Fig. 2, the entries of vector $\boldsymbol{y}$ (i.e., D) correspond to positive and negative vertical positions of the blue circles calculated at 7 time points $[0.2, 0.3, \ldots, 0.8]^T$ defined in a normalized unit interval, and the horizontal position of the green circles associated to the jitter, respectively. The overall number of output components is $D = 16$, while the number of input parameters $p = 11$. Within the machine learning framework, the modeling problem stated previously can be reformulated as the problem of building a generic vector-valued surrogate model $\boldsymbol{\mathcal{M}} : \mathcal{X} \rightarrow \mathcal{Y}$, starting from the information available on the training set $\mathcal{D} = \{(\boldsymbol{x}_l, \boldsymbol{y}_l)\}_{l=1}^L$, where L represents the total number of training pairs, such that $\boldsymbol{x}_l \in \mathcal{X}$ corresponds to the lth configuration of the input parameters and $\boldsymbol{y}_l = [y_l^{(1)}, \ldots, y_l^{(D)}]^T \in \mathcal{Y}$ is the corresponding output vector.

B. Scalar-Output SVM and LS-SVM Regressions

The abovementioned vector-valued learning scenario can be reinterpreted as the problem of learning D independent scalar regressions (i.e., one for each output component), each of them considering the training set $\mathcal{D}^{(i)} = \{(\boldsymbol{x}_l, y_l^{(i)})\}_{l=1}^L$, such that $\mathcal{D} = \cup_{i=1}^D \mathcal{D}^{(i)}$, in which $y_l^{(i)}$ is ith component of the output vector associated to the lth input configuration $\boldsymbol{x}_l$. This allows to reformulate the overall vector-valued learning problem in terms of D independent scalar-output regressions, which can be suitably tackled with standard kernel machine approaches, such as the SVM and LS-SVM regressions [4].

Specifically, each training set $\mathcal{D}^{(i)} = \{(\boldsymbol{x}_l, y_l^{(i)})\}_{l=1}^L$ can be used together with the SVM and LS-SVM regression to train the corresponding surrogate model [15], [16], such that

$$\mathcal{M}^{(i)}_{\text{SVM/LS-SVM}}(\boldsymbol{x}) = \sum_{l=1}^{L} \beta_l^{(i)} k^{(i)}(\boldsymbol{x}_l, \boldsymbol{x}) + b^{(i)} \tag{1}$$

where $\beta_l^{(i)}$ and $b^{(i)}$ are the regression coefficients and bias term estimated during the model training and $k^{(i)}(\cdot,\cdot) : \mathbb{R}^{p\times p} \rightarrow \mathbb{R}$ is a scalar kernel function associated to the ith output component.

Several kernel functions $k(\cdot,\cdot)$ have been proposed in the literature. Hereafter in this article, we will adopt the Gaussian radial basis function (RBF) kernel, which writes [15]

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|^2}{2\sigma^2}\right) \tag{2}$$

where σ^2 is the kernel hyperparameter.

The SVM and the LS-SVM regression share the same dual space representation in (1). However, for the SVM regression the coefficients $\beta_l^{(i)}$ and the bias terms $b_l^{(i)}$ are estimated by minimizing the ε-insensitive loss function computed between the model predictions and training outputs, while for the LS-SVM regression the abovementioned unknowns are estimated as the ones minimizing a squared loss function [15], [16]. A Tikhonov regularizer and its corresponding hyperpameter are used by both methods to suppress overfitting and to improve their generalization on the test set, by minimizing the model variance and thus its sensitivity to the training set [40].

Concerning the training algorithms, the SVM regression is built by numerically solving a quadratic optimization problem [15], and thanks to the ε-insensitive norm, it can lead to a sparse solution in which some of the coefficients $\beta_l^{(i)} = 0$. On the other hand, the training of the LS-SVM regression can be done in a closed-form as the solution of a linear system of equations. Concerning the model hyperparameters, the SVM regression with Gaussian RBF kernel requires the tuning of three hyperparameters: the shape factor σ in (2), the width of the ε-insensitive zone and the regularizer hyperparameters. On the other hand, the LS-SVM regression with Gaussian RBF kernel requires the tuning of two hyperparameters: the shape factor σ in (2) and the regularizer hyperparameters. For each regression the abovementioned hyperparameters are tuned via cross-validation [40]. The statistics and machine learning toolbox and the LS-SVMLab toolbox version 1.8 [41] have been used for the SVM and the LS-SVM regression, respectively.

It is important to remark that this type of modeling scheme for vector-valued problem, based on single-output regressions, requires the training and tuning of the hyperparameters of D regression models, thus compromising its efficiency in regression problems with a large number of output components [42].

C. Vector-Valued KRR

An alternative modeling approach consists of applying a vector-valued formulation of the KRR, which allows to directly account for the vector-valued nature of the regression problem at hand [6], [7], [17]. Specifically, the vector-valued KRR can be directly adopted to train a single vector-valued surrogate model $\boldsymbol{\mathcal{M}}_{\text{KRR}} : \mathcal{X} \rightarrow \mathcal{Y}$, which according to the represented theorem for vector-valued regression problems presented in [44], writes

$$\boldsymbol{\mathcal{M}}_{\text{KRR}}(\boldsymbol{x}) = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)\boldsymbol{c}_l \tag{3}$$

where $\boldsymbol{\mathcal{M}}_{\text{KRR}}(\boldsymbol{x}) = [\mathcal{M}^{(1)}_{\text{KRR}}(\boldsymbol{x}), \ldots, \mathcal{M}^{(D)}_{\text{KRR}}(\boldsymbol{x})]^T$ is a vector collecting the model prediction for any $\boldsymbol{x} \in \mathcal{X}$, $\mathbf{K}(\cdot,\cdot) : \mathbb{R}^{p\times p} \rightarrow \mathbb{R}^{D\times D}$ is a multioutput kernel matrix and

$\boldsymbol{c}_l = [c_{1,l}, \ldots, c_{D,l}]^T \in \mathbb{R}^D$ are column vectors collecting the regression unknowns.

A separable structure consisting of the product of scalar kernel function is considered for the matrix kernel function $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')$ [17], such that

$$[\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')]_{[d,d']} = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}')k_o(d, d') \tag{4}$$

where $k_{\boldsymbol{x}}$ and k_o are scalar kernels acting independently on the input space (i.e., $k_{\boldsymbol{x}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$) and on the output dimensions (i.e., $k_o : \{1, \ldots, D\} \times \{1, \ldots, D\} \to \mathbb{R}$). Without loss of generality, the Gaussian RBF kernel in (2) will be used hereafter for the kernel $k_{\boldsymbol{x}}$ and k_o, respectively.

For the considered model structure, the regression coefficients collected in the vectors $\boldsymbol{c}_l$ can be estimated in a closed-form via the solution of a discrete-time Sylvester equation (additional mathematical details are provided in [6]), which writes

$$\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + \lambda\mathbf{C} = \mathbf{Y} \tag{5}$$

where λ is the hyperparameter associated to the Tikhonov regularizer, $\mathbf{K}_{\boldsymbol{x}}$ is a $L \times L$ Gram matrix computed from the input samples $\{\boldsymbol{x}_l\}_{l=1}^L$ (i.e., $[\mathbf{K}_{\boldsymbol{x}}]_{ij} = k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$), $\mathbf{B}$ is a $D \times D$ Gram matrix computed on the output dimensions $\{1, \ldots, D\}$ (i.e., $[\mathbf{B}]_{ij} = k_o(d_i, d_j)$), $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L]^T \in \mathbb{R}^{L \times D}$ is a matrix collecting the model unknowns and $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l]^T$ is a $L \times D$ matrix associated to the training output.

Thanks to the properties of the Kronecker product, the above-mentioned matrix equation can be vectorized and reformulated in terms of a standard linear system with (LD) equations in (LD) unknowns [43] with a computational cost proportional to $\mathcal{O}(L^3D^3)$ for a plain inversion or to $\mathcal{O}(KL^2D^2)$, where K is the number of iteration, for a conjugate gradient descent algorithm [6]. On the other hand, this article will use a more efficient implementation based on a diagonalization procedure, inspired by [45], for which the computational cost for the model training reduces from $\mathcal{O}(L^3D^3)$ to $\mathcal{O}(L^3 + D^3 + L^2D + LD^2)$, thus leading to beneficial effect on the training time when the product (LD) is large (additional details can be found in [46]).

Different from modeling strategies based on scalar-output regressions in Section IV-B, the training of the proposed implementation of the vector-valued KRR requires the tuning of three hyperparameters only. Such tuning is done via a 3-fold cross validation. Moreover, the vector-valued formulation allows to learn possible correlations in both the input and output dimensions, thus improving the model reliability and robustness to noise.

V. Numerical Results

The simulation setup described in Section III was used to generate the training and test sets used to construct and assess the accuracy of a surrogate models constructed via the SVM, LS-SVM, and vector-valued KRR regressions presented in Section IV. These training and test sets were generated by randomly varying the considered 11 parameters defining the link, according to the range of variation discussed in Section II, via a Latin hypercube sampling (LHS) scheme [47]. For each set of parameters, the received differential voltage $v_d(t)$ was recorded and the eye aperture was computed according to the procedure in Section III. All performance figures in the present section were computed using 5000 test samples, i.e., 5000 different responses of the received voltage v_d along with its corresponding eye plot and inner polygon. Also in this case, the input configurations of the test were generated by randomly varying the considered 11 parameters of the link according to the range of variation discussed in Section II via an LHS scheme.

TABLE I
Absolute Average Error Computed From the Predictions of the Surrogate Models Built Via the Regression Techniques Presented in Section IV for an Increasing Number Training Sample (i.e., $L = 60$, 150, and 300) on a Test Set With 5000 Eye Realizations

	Methods	SVM	LS-SVM	Vector-valued KRR
L	pairs	Ave error	Ave error	Ave error
60	1_u–1_d (mV)	6.21	6.18	**5.97**
	2_u–2_d (mV)	7.49	7.25	**6.73**
	3_u–3_d (mV)	**6.65**	6.68	6.91
	4_u–4_d (mV)	7.49	**7.17**	7.29
	5_u–5_d (mV)	10.13	6.80	**6.68**
	6_u–6_d (mV)	4.03	3.83	**3.79**
	7_u–7_d (mV)	4.25	4.07	**4.03**
	$8_r - 8_l$ (ps)	3.62	**3.22**	3.34
150	1_u–1_d (mV)	4.73	4.61	**4.52**
	2_u–2_d (mV)	**4.16**	4.27	4.20
	3_u–3_d (mV)	4.93	**4.92**	4.99
	4_u–4_d (mV)	9.99	4.86	**4.75**
	5_u–5_d (mV)	4.22	**3.78**	3.79
	6_u–6_d (mV)	3.06	**2.72**	2.73
	7_u–7_d (mV)	2.72	2.66	**2.65**
	$8_r - 8_l$ (ps)	2.67	**2.63**	2.67
300	1_u–1_d (mV)	3.50	**3.44**	3.51
	2_u–2_d (mV)	3.25	3.19	**3.13**
	3_u–3_d (mV)	3.75	**3.71**	3.72
	4_u–4_d (mV)	4.05	**3.96**	4.03
	5_u–5_d (mV)	3.08	**2.96**	2.99
	6_u–6_d (mV)	2.38	2.40	**2.33**
	7_u–7_d (mV)	2.32	2.31	**2.20**
	$8_r - 8_l$ (ps)	2.23	2.15	**2.12**

Bold text is used to highlight the lowest errors among the different modeling methods.

Table I provides a quantitative picture of the accuracy of the considered three surrogate modeling approaches for an increasing number of training samples L from 60 to 300 (see first column). Both the eye apertures in mV, computed at the different location pairs labeled from one to seven in Fig. 2, and the eye width in ps corresponding to the eighth pair in the same figure are considered (second column). The remaining columns collect the

average absolute errors computed as the mean absolute deviation of the eye apertures in mV and eye width in ps predicted by the considered surrogate models, and the corresponding ones obtained from the 5000 reference eye realizations in the test set. Bold text is used to highlight the lowest errors among the different modeling methods. This comparison clearly highlights the capability of all the considered surrogate models to learn the actual information provided by the training set. Among them, in most of the considered modeling scenarios, the vector-valued KRR shows an improved accuracy compared to the SVM and LS-SVM regression, thanks to its capability of learning possible correlation among the output components.

Moreover, as shown in Fig. 3, the vector-valued KRR approach also offers a faster training time with respect to the other plain scalar-output kernel regressions, since its training CPU time seems to be less affected by the number of training samples (i.e., L). It is important to remark that the CPU times reported in Fig. 3 refer to the training time only, and do not include the computational cost relative to the generation of the training set.

The generation of 300 training samples required approximately 11 h. This expensive data-collection phase must be carried out once. After the model training, its evaluation on the 5000 test samples required few seconds, while the corresponding SPICE simulations required four days.

Fig. 4 shows all the possible eye apertures computed for the test samples (see gray cloud) superimposed by the best, median, and worst profiles of the inner area computed using the reference responses (top panel, black lines) and, respectively, the proposed vector-valued KRR surrogate model (bottom panel, blue lines). The median curve corresponds to the 50% threshold between all the larger and the smaller eye apertures. This additional and less conventional performance figure stresses the flexibility of the proposed approach in possibly providing the information of the probability (e.g., in terms of a quantiles) associated to a given eye aperture.

Similarly, Fig. 5 plots three possible eye apertures defined by their corresponding polygons associated to three sets of uncertain parameters. The figure compares the prediction obtained using the proposed vector-valued KRR model trained with 300 training samples with the reference profiles, thus highlighting the very good accuracy of the proposed surrogate modeling approach.

As an additional and final validation, the eye quality is assessed by a single scalar quantity, such as the eye aperture computed from the proposed polygonal approximation of the inner-eye area. For the same test case, Fig. 6 compares the probability density functions (PDFs) computed from the same 5000-sample test set, used as reference, with the corresponding ones obtained from the surrogate model built via the LS-SVM regression, the SVM regression and the proposed vector-valued KRR by using $L = 300$ samples. The three panels show the PDFs associated to the eye area (left panel), eye height computed in the region of the fourth polygonal edge pairs of Fig. 2, i.e., $(4_u) - (4_d)$ (central panel) and eye width, again defined between pairs $(8_r) - (8_l)$ (right panel). The eye area is computed as briefly outlined in the fourth item of the procedure described in the appendix. The

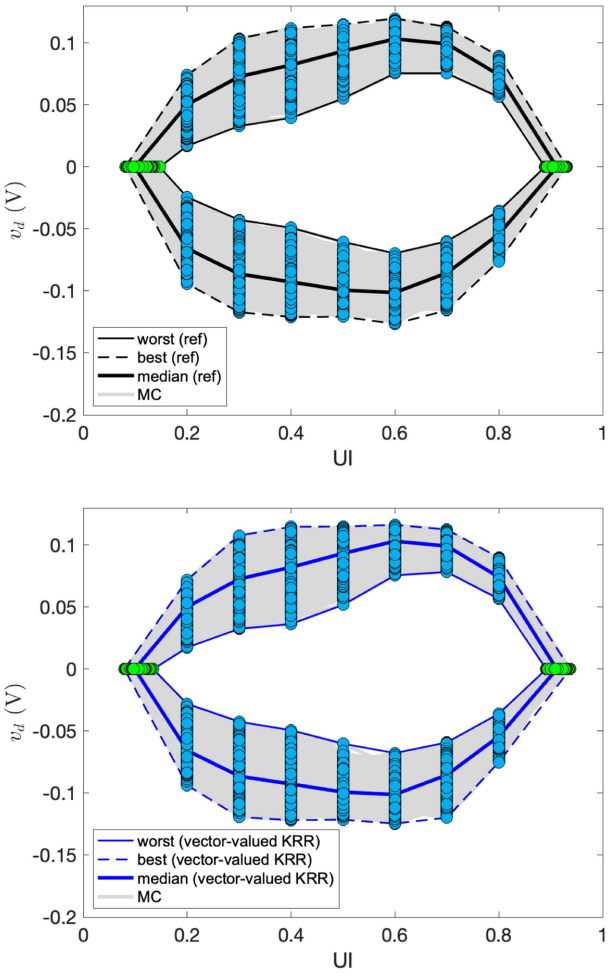

Fig. 4. Data-link analysis carried out by computing the best, the worst (dashed and solid thin lines), and the median (thick) inner-eye aperture. The above-mentioned eye apertures are superimposed to the complete set of eye diagrams obtained for the test set (light-gray cloud). The blue and the green clouds of circles play the same role of the corresponding colored circles in Fig. 2. Top panel: reference; bottom panel: vector-valued KRR.

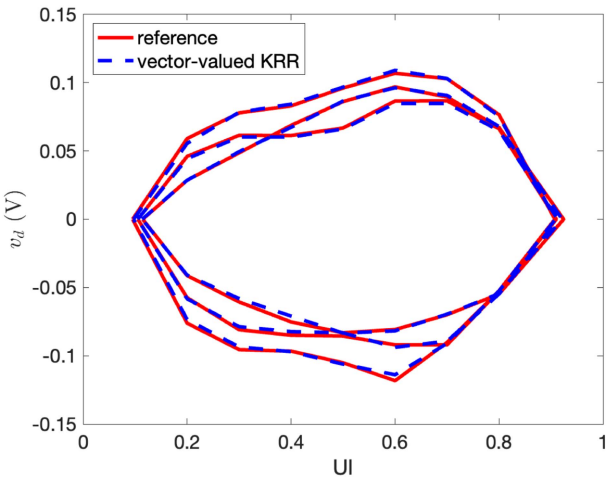

Fig. 5. Eye aperture computed for three sets of uncertain data-link parameters. The eye opening polygons obtained by the proposed vector-valued KRR model are compared to the reference ones.

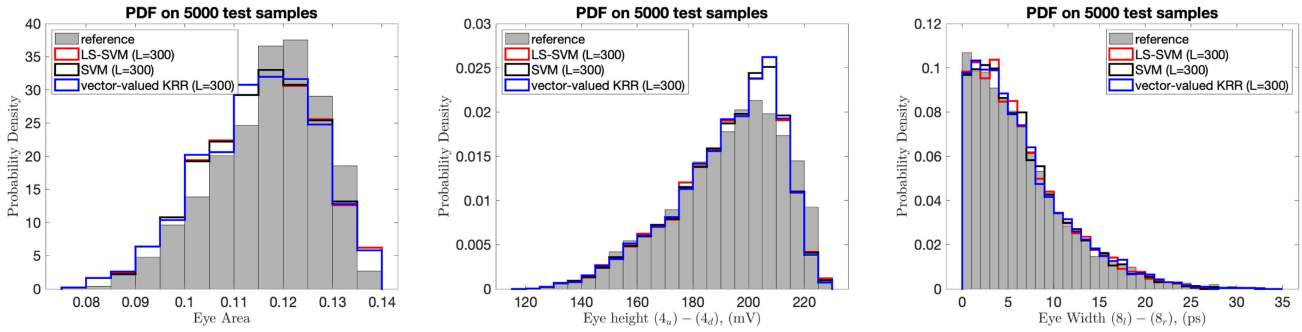

Fig. 6. Comparison between the PDFs of the eye area, height, and width computed from the predictions of a surrogate model built via the LS-SVM regression, the LS-SVM regression and the vector-valued KRR with $L = 300$ training samples and the corresponding ones calculated on 5000 test samples.

abovementioned comparison highlights a good agreement among the PDFs predicted by the proposed surrogate models and the ones computed from 5000 test samples, especially on the tails of the distributions.

VI. Conclusion

The main novelty of this article is the stochastic time-domain mapping approach. For a given eye pattern, the inner eye is mapped using a suitable polygonal approximation. After an initial training stage, a parametric surrogate model is built, predicting the effect of system uncertainties on the shape of this polygon. This allows designers to effectively compute the metric of their choice or use eye masks in a stochastic context. The procedure was demonstrated in the case of a flexible data link for wearable applications, with uncertain electrical and geometrical parameters, using machine learning regression to build the surrogate model. Data were collected by simulating the link in a differential configuration operating at 2 Gbps. Three types of regressions were used, showing that for the problem at hand, a vector formulation of the so-called KRR is the preferable choice. A nice additional feature is that the latter naturally embeds the possible correlation, which exists among the vertices of the eye polygon. This also has a beneficial effect in terms of training time, model reliability, and robustness.

Stochastic mapping takes variability analysis one step further. It is of particular interest for applications where either the level of uncertainties or their impact is particularly high. These include flexible or structural electronics, ultralow power systems, systems designed to operate in various hostile environments, but also very-low cost applications where high tolerances are a consequence of financial considerations.

Appendix

The proposed algorithm for the computation of the eye aperture through a polynomial approximation is described in the following. The procedure is outlined, including the essential steps using a metalanguage description and a MATLAB-like code. It is assumed that the sampled response of $v_d(t)$ is stored in the vector `vd`. It collects the values of the received differential voltage at the time samples stored in vector `t`. Also, due to the differential communication scheme, a zero-threshold (`threshold = 0` in the code) is assumed for computing the state (i.e., bit) transitions.

The main steps of the algorithm are as follows.

1) The estimated crossing times are store in vector `tz` and are determined via linear interpolation

```
>> vd = vd-threshold
>> iz = find(diff(sign(vd)))
>> a  = (vd(iz)-vd(iz-1))./(t(iz)
        -t(iz-1))
>> tz = (-vd(iz)+a.*t(iz))./a.
```

2) The time axis and the crossing times are wrapped into the unit interval defined by the bit time (`TBIT`); also, the jitter width (`JITW`) is computed by considering the largest and the smallest wrapped (i.e., normalized) transition times

```
>> tzoffset = mod(tz(1),TBIT)
>> tzw      = mod(tz-tzoffset,TBIT)
>> JLEFT  = TBIT-min(tzw(tzw>TBIT/2))
>> JRIGHT   = max(tzw(tzw<TBIT/2))
>> JITW     = JLEFT+JRIGHT.
```

3) Within the unit interval, a number of sampling points is set (e.g., `POINTS = [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]`) and the minimum and the maximum values of the received signal observed at these sampling points are computed, $v_d(t = (\texttt{POINTS(kk)} + (\texttt{n} - 1)) * \texttt{TBIT})$. In the abovementioned notation, the index `kk` varies from one to the max number of items in vector `POINTS`, and `n` ranges from one to the maximum number of bits composing the bitstream. The procedure in this step is suitably modified to avoid probing the differential signal in the jitter region. It is important to point out that the abovementioned max and min values define, for a given point in the unit interval, the position of the vertical eye height (e.g., see the blue circles in Fig. 2). Also, the green circles in on the left and on the right side of the eye correspond, in the x-axis, to `JRIGHT/TBIT` and `1-JLEFT/TBIT`, respectively.

4) Based on the inner polygonal shape built in the previous steps, different eye parameters (such as the eye width, height, and area) can be readily computed. Among the previous parameters, the area can be computed via embedded functions, such as `polyarea` in MATLAB or via the superposition of the 2-D basic shapes (i.e., triangles or trapezoids) defined by the different vertices of the inner-eye polygon, such as the one shown if Fig. 2.

REFERENCES

[1] R. Spence and R. S. Soin, *Tolerance Design of Electronic Circuits*. London, U.K.: Imperial College Press, 1997.

[2] Q. Zhang, J. J. Liou, J. McMacken, J. Thomson, and P. Layman, "Development of robust interconnect model based on design of experiments and multiobjective optimization," *IEEE Trans. Electron Dev.*, vol. 48, no. 9, pp. 1885–1891, Sep. 2001.

[3] P. Manfredi, D. V. Ginste, I. S. Stievano, D. De Zutter, and F. G. Canavero, "Stochastic transmission line analysis via polynomial chaos methods: An overview," *IEEE Electromagn. Compat. Mag.*, vol. 6, no. 3, pp. 77–84, Jul.–Sep. 2017.

[4] R. Trinchero, M. Larbi, H. Torun, F. G. Canavero, and M. Swaminathan, "Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2019.

[5] R. Trinchero and F. G. Canavero, "Modeling of eye diagram height in high-speed links via support vector machine," in *Proc. IEEE 22nd Workshop Signal Power Integrity*, Brest, France, 2018, pp. 1–4.

[6] N. Soleimani and R. Trinchero, "Compressed complex-valued least squares support vector machine regression for modeling of the frequency-domain responses of electromagnetic structures," *Electronics*, vol. 11, no. 4, 2022, Art. no. 551.

[7] N. Soleimani, R. Trinchero, and F. G. Canavero, "Bridging the gap between artificial neural networks and kernel regressions for vector-valued problems in microwave applications," *IEEE Trans. Microw. Theory Techn.*, vol. 71, no. 6, pp. 2319–2332, Jun. 2023.

[8] T. Nguyen et al., "Comparative study of surrogate modeling methods for signal integrity and microwave circuit applications," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 11, no. 9, pp. 1369–1379, Sep. 2021.

[9] H. Ma et al., "A fast optimization method for high-speed link inverse design with SVR-AS algorithm," *IEEE Trans. Signal Power Integrity*, vol. 1, pp. 22–31, 2022.

[10] S. Kushwaha et al., "Comparative analysis of prior knowledge-based machine learning metamodels for modeling hybrid Copper–Graphene on chip interconnects," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 6, pp. 2249–2260, Dec. 2022.

[11] K. Dimple et al., "Modified knowledge-based neural networks using control variates for the fast uncertainty quantification of on-chip MWCNT interconnects," *IEEE Trans. Electromagn. Compat.*, vol. 65, no. 4, pp. 1232–1246, Aug. 2023, doi: 10.1109/TEMC.2023.3279695.

[12] R. Kumar et al., "Knowledge-based neural networks for fast design space exploration of hybrid copper-graphene on-chip interconnect networks," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 1, pp. 182–195, Feb. 2022.

[13] S. Han, O. W. Bhatti, and M. Swaminathan, "A knowledge based method for optimization of decoupling capacitors in power delivery networks," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 12, no. 5, pp. 828–838, May 2022.

[14] H. Ma et al., "Uncertainty quantification of signal integrity analysis for neuromorphic chips," *IEEE Trans. Signal Power Integrity*, vol. 1, pp. 160–169, 2022.

[15] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Berlin, Germany: Springer, 1999.

[16] J. A. K. Suykens et al. *Least Squares Support Vector Machines*. Singapore: World Scientific Pub Co Inc, 2002.

[17] A. Mauricio, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, 2012.

[18] A. Rudi, L. Carratino, and L. Rosasco, "FALKON: An optimal large scale kernel method," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3891–3901.

[19] D. Cottet, J. Grzyb, T. Kirstein, and G. Troster, "Electrical characterization of textile transmission lines," *IEEE Trans. Adv. Packag.*, vol. 26, no. 2, pp. 182–190, May 2003.

[20] D. Marculescu et al., "Electronic textiles: A platform for pervasive computing," *Proc. IEEE*, vol. 91, no. 12, pp. 1995–2018, Dec. 2003.

[21] R. Nayak, L. Wang, and R. Padhye, "Electronic textiles for military personnel," in *Electronic Textiles*, T. Dias, Ed., Sawston, U.K.: Woodhead Publishing, 2015, pp. 239–256.

[22] J. S. Chang, A. F. Facchetti, and R. Reuss, "A circuits and systems perspective of organic/printed electronics: Review, challenges, and contemporary and emerging design approaches," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 1, pp. 7–26, Mar. 2017.

[23] T. Agcayazi, K. Chatterjee, A. Bozkurt, and T. K. Ghosh, "Flexible interconnects for electronic textiles," *Adv. Mater. Technol.*, vol. 3, no. 10, Oct. 2018, Art. no. 1700277.

[24] J. Shi et al., "Smart textile-integrated microelectronic systems for wearable applications," *Adv. Mater.*, vol. 32, no. 5, Feb. 2020, Art. no. 1901958.

[25] M. Wagih et al., "Microwave-enabled wearables: Underpinning technologies, integration platforms, and next-generation roadmap," *IEEE J. Microw.*, vol. 3, no. 1, pp. 193–226, Jan. 2023.

[26] "LTspice information center analog devices," Linear Technology. Accessed: Sep. 26, 2023. [Online]. Available https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html

[27] M. Telescu, R. Trinchero, I. S. Stievano, and N. Tanguy, "Worst-case optimization of a digital link for wearable electronics in a stochastic framework," in *Proc. IEEE 26th Workshop Signal Power Integrity*, Siegen, Germany, 2022, pp. 1–4.

[28] C. R. Paul, *Analysis of Multiconductor Transmission Lines*. Hoboken, NJ, USA: Wiley, 2007.

[29] C. R. Paul and A. E. Feather, "Computation of the transmission line inductance and capacitance matrices from the generalized capacitance matrix," *IEEE Trans. Electromagn. Compat.*, vol. 18, no. 4, pp. 175–183, Nov. 1976.

[30] M. Dyson and T. Skyrme, "E-textiles and smart clothing markets 2023-2033: Technologies, players, and applications," Accessed: Sep. 26, 2023. [Online]. Available: https://www.idtechex.com/en/research-report/e-textiles-and-smart-clothing-markets-2023-2033-technologies-players-and-applications/925

[31] C. Siviero, R. Trinchero, S. Grivet-Talocia, I. S. Stievano, G. Signorini, and M. Telescu, "Constructive signal approximations for fast transient simulation of coupled channels," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 9, no. 10, pp. 2087–2096, Oct. 2019.

[32] G. Signorini, C. Siviero, M. Telescu, and I. S. Stievano, "Present and future of I/O-buffer behavioral macromodels," *IEEE Electromagn. Compat. Mag.*, vol. 5, no 3, pp. 79–85, Jul.–Sep. 2016.

[33] I. S. Stievano, I. A. Maio, F. G. Canavero, and C. Siviero, "Reliable eye-diagram analysis of data links via device macromodels," *IEEE Trans. Adv. Packag.*, vol. 29, no 1, pp. 31–38, Feb. 2006.

[34] T. Reuschel et al., "Efficient prediction of equalization effort and channel performance for PCB-Based data links," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 7, no. 11, pp. 1842–1851, Nov. 2017.

[35] K. Cho et al., "Signal integrity design and analysis of differential high-speed serial links in silicon interposer with through-silicon via," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 9, no. 1, pp. 107–121, Jan. 2019.

[36] D. Oh, J. Ren, and S. Chang, "Hybrid statistical link simulation technique," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 1, no. 5, pp. 772–783, May 2011.

[37] D. Kim, H. Kim, and Y. Eo, "Analytical eye-diagram determination for the efficient and accurate signal integrity verification of single interconnect lines," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 10, pp. 1536–1545, Oct. 2012.

[38] W. T. Beyene, "Application of artificial neural networks to statistical analysis and nonlinear modeling of high-speed interconnect systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 1, pp. 166–176, Jan. 2007.

[39] N. Ambasana, G. Anand, B. Mutnury, and D. Gope, "Eye height/width prediction from *S*-parameters using learning-based models," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 6, no. 6, pp. 873–885, Jun. 2016.

[40] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial," 2019, *arXiv:1905.12787*.

[41] "LS-SVMlab, version 1.8," Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven: Leuven, Belgium, 2011. Available online: http://www.esat.kuleuven.be/sista/lssvmlab/

[42] P. Manfredi and R. Trinchero, "A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses," *IEEE Access*, vol. 8, pp. 92019–92027, 2020.

[43] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[44] C. A. Micchelli and M. Pontil, "On learning vector–valued functions," *Neural Comput.*, vol. 17, pp. 177–204, 2005.

[45] L. Baldassarre et al., "Multi-output learning via spectral filtering," *Mach. Learn.*, vol. 87, pp. 259–301, 2012.

[46] N. Soleimani, P. Manfredi, and R. Trinchero, "Efficient implementation of the vector-valued kernel ridge regression for the uncertainty quantification of the scattering parameters of a 2-GHz low-noise amplifier," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Model. Optim.*, Winnipeg, MB, Canada, 2023, pp. 143–146.

[47] M. McKay, R. Beckman, and W. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

Mihai Telescu (Member, IEEE) received the engineer's degree in electronics and telecommunications from the Polytechnic University of Timisoara, Timisoara, Romania, in 2003, the M.Sc. degree from the University of Rennes 1, Rennes, France, in 2004, and the doctorate degree in electronics from the University of Brest (UBO), Brest, France, in 2007.

He was a Postdoc Researcher with the EMC Group, Politecnico di Torino, Torino, Italy, between 2008 and 2009. Since then, he has been an Associate Professor with UBO. His research interests include linear and nonlinear macromodeling, model order reduction, system identification, stochastic analysis, and related topics.

Dr. Telescu chaired the 22nd and the 23rd editions of the IEEE Workshop on Signal and Power Integrity and is an Associate Editor for IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY.

Riccardo Trinchero (Member, IEEE) received the M.Sc. and Ph.D. degrees in electronics and telecommunications engineering from Politecnico di Torino, Torino, Italy, in 2011 and 2015, respectively.

He is currently an Associate Professor with the EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino. His research interests include the analysis of switching dc–dc converters, machine learning, and statistical simulation of circuits and systems.

Nastaran Soleimani (Student Member, IEEE) received the M.Sc. degree in electrical engineering electronics from the Ferdowsi University of Mashhad, Mashhad, Iran, in 2016, and the Ph.D. degree in electronics and telecommunications engineering from Politecnico di Torino, Torino, Italy, in 2023.

She is currently a Postdoctoral Researcher with the Department of Electronics and Telecommunications, Politecnico di Torino. Her research interests include machine learning and statistical simulation of circuits and systems.

Noel Tanguy received the M.Sc. and the Ph.D. degrees (Thèse de Doctorat) in electronics from the University of Brest (UBO), Brest, France, in 1989, and 1994, respectively, where he received the HDR degree (accreditation to supervise research) in 2006.

He was an Associate Professor between 1995 and 2008 and has been a Full Professor with UBO ever since. He currently conducts his research with the Lab-STICC (Laboratoire des sciences et technologies de l'information, de la communication et de la connaissance, UMR CNRS). His research interests include signal integrity, modeling, and approximation theory, model order reduction, linear and nonlinear macromodeling, impairments estimation and compensation techniques in optical communications, stochastic analysis, and related topics.

Igor Simone Stievano (Senior Member, IEEE) received the master's degree in electronic engineering and the Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Torino, Italy, in 1996 and 2001, respectively.

He is currently a Professor of electrical engineering with the Department of Electronics and Telecommunications, Politecnico di Torino. From 2017 to 2021, he was the Vice Rector of academic and scientific activities with the joint campus of the Politecnico di Torino in Uzbekistan, Turin Polytechnic University in Tashkent. He has authored or coauthored more than 130 papers published in international journals and conference proceedings. His research interests include electromagnetic compatibility and signal integrity, with emphasis on modeling and simulation of digital circuits, transmission lines, PLC channels, switching converters, development of stochastic methods for the statistical simulation of circuits and systems, and the compact modeling of electrical and gas networks via a complex network paradigm and simplified graph-based approaches.

Dr. Stievano was the Program Co-Chair of the 20th and 21st IEEE Workshops on Signal and Power Integrity (SPI2016 and SPI2017).